Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

Claim 1 (currently amended): A dictionary comprising codevectors of variable dimension and intended to be used in a device for compression coding and/or decoding of digital signals, by vector quantization at variable rate defining a variable resolution, wherein ~~characterized in that it comprises~~:

- on the one hand, for a given dimension, inter-embedded dictionaries of increasing resolution,
- and, on the other hand, for a given dimension, a union:

- of a first set consisting of codevectors constructed by inserting, into codevectors of dictionaries of lower dimension, elements taken from a finite set of real numbers according to a finite collection of predetermined insertion rules,
- and of a second set consisting of codevectors that may not be obtained by insertion into codevectors of lower dimension of the elements of said finite set according to said collection of insertion rules.

Claim 2 (currently amended): The dictionary as claimed in claim 1, wherein ~~characterized in that~~ said collection of insertion rules is formulated on the basis of elementary rules consisting in inserting a single element of the finite set of real numbers in the guise of component at a given position of a vector.

Claim 3 (currently amended): The dictionary as claimed in claim 2, wherein ~~characterized in that~~ each elementary rule is defined by a pair of two positive integers representative:

- of a rank of the element in said finite set,
- and of a position of insertion.

Claim 4 (currently amended): A method for forming a dictionary ~~according to one of claims 1 to 3, the dictionary~~ comprising codevectors of variable dimension and intended to be used in a device for compression coding and/or decoding of digital signals, by vector quantization at variable rate defining a variable resolution,

in which, for a given dimension:

a) a first set consisting of codevectors formed by inserting/deleting into/from codevectors of dictionaries of lower/higher dimension elements taken from a finite set of real numbers according to a finite collection of predetermined insertion/deletion rules,

b) a first, intermediate, dictionary comprising at least said first set is constructed, for said given dimension,

c) and, to adapt said dictionary to a use with at least one given resolution, a second, definitive, dictionary is constructed, on the basis of the intermediate dictionary, by embedding/simplification of dictionaries of increasing/decreasing resolutions, the dictionaries of increasing resolutions being inter-embedded from the dictionary of smallest resolution up to the dictionary of greatest resolution.

Claim 5 (original): The method as claimed in claim 4, in which, for a given dimension **N**:

a0) an initial dictionary of initial dimension **n**, lower than said given dimension **N**, is obtained,

a1) a first set consisting of codevectors of dimension **n+i** formed by inserting into codevectors of the initial dictionary elements taken from a finite set of real numbers according to a finite collection of predetermined insertion rules is constructed,

a2) there is provided a second set consisting of codevectors of dimension **n+i** that may not be obtained by insertion into the codevectors of the initial dictionary of the elements of said finite set with said collection of insertion rules,

a3) an intermediate dictionary, of dimension **n+i** comprising a union of said first set and of said second set is constructed,

and steps a1) to a3) are repeated, at most **N-n-1** times, with said intermediate dictionary in the guise of initial dictionary, up to said given dimension **N**.

Claim 6 (original): The method as claimed in claim 4, in which, for a given dimension **N**:

a'0) an initial dictionary of initial dimension **n**, higher than said given dimension **N**, is obtained,

a'1) a first set, of dimension **n-i**, is constructed by selection and extraction of possible codevectors of dimension **n-i** from the dictionary of dimension **n**, according to a finite collection of predetermined deletion rules,

a'2) there is provided a second set consisting of codevectors of dimension **n-i**, that may not be obtained by deletion, from the codevectors of the initial dictionary, of the elements of said finite set with said collection of deletion rules,

a'3) an intermediate dictionary, of dimension **n-i** comprising a union of said first set and of said second set is constructed,

and steps a'1) to a'3) are repeated, at most **n-N-1** times, with said intermediate dictionary in the guise of initial dictionary, up to said given dimension **N.**

Claim 7 (currently amended): The method as claimed in claim 5 ~~claims 5 and 6~~, in which **N** successive dictionaries of respective dimensions **1** to **N** are obtained on the basis of an initial dictionary of dimension **n**, through the repeated implementation of steps a1) to a3) for the dimensions **n+1** to **N**, and through the repeated implementation of steps a'1) to a'3) for the dimensions **n-1** to **1.**

Claim 8 (currently amended): The method as claimed in claim 4 ~~one of claims 4 to 7~~, in which said collection of insertion/deletion rules is formulated on the basis of elementary rules consisting in inserting/deleting a single element of the finite set of reals in the guise of component at a given position of a vector.

Claim 9 (original): The method as claimed in claim 8, in which each elementary rule is defined by a pair of two positive integers representative:

- of a rank of the element in said finite set,
- and of a position of insertion/deletion.

Claim 10 (currently amended): The method as claimed in claim 4 ~~one of claims 4 to 9~~, in which said finite set and said collection of insertion/deletion rules are defined a priori, before constructing the dictionary by analysis of a source to be quantized.

Claim 11 (original): The method as claimed in claim 10, in which said source is modelled by a learning sequence and the definition of said finite set and of said collection of insertion/deletion rules is effected by statistical analysis of said source.

Claim 12 (currently amended): The method as claimed in claim 10 ~~one of claims 10 and 11~~, in which said finite set is chosen by estimation of a monodimensional probability density of said source.

Claim 13 (currently amended): The method as claimed in claim 4 ~~one of claims 4 to 9~~, in which said finite set and said collection of insertion/deletion rules are defined a posteriori after construction of dictionaries by embedding/simplification of dictionaries of successive resolutions, followed by a statistical analysis of these dictionaries thus constructed.

Claim 14 (currently amended): The method as claimed in claim 10 ~~claims 10 and 13~~, in which:

- a first set and a first collection of insertion/deletion rules are chosen a priori by analysis of a learning sequence, so as to form one or more intermediate dictionaries,
- at least one part of said first set and/or of said first collection of insertion/deletion rules is updated by a posteriori analysis of said one or more intermediate dictionaries,
- and, as appropriate, at least one part of the set of codevectors forming said one or more intermediate dictionaries is also updated.

Claim 15 (currently amended): The method as claimed in claim 4 ~~one of claims 4 to 14~~, in which step c) comprises the following operations:

c0) an initial dictionary of initial resolution $\mathbf{r}_n$, lower than said given resolution $\mathbf{r}_N$, is obtained,

c1) on the basis of the initial dictionary, an intermediate dictionary of resolution $\mathbf{r}_{n+1}$ higher than the initial resolution $\mathbf{r}_n$ is constructed,

c2) operation c1) is repeated until the given resolution $\mathbf{r}_N$ is attained.

Claim 16 (original): The method as claimed in claim 15, in which, for each iteration of operation c1), there is provided a construction of classes and of centroids, in which the centroids belonging at least to the dictionaries of resolution higher than a current resolution $\mathbf{r}_i$ are recalculated and updated.

Claim 17 (original): The method as claimed in claim 16, in which the centroids which belong to the dictionaries of resolution lower than a current resolution $\mathbf{r}_i$ are updated only if the total distortions of all the dictionaries of lower resolution are decreasing from one update to the next.

Claim 18 (currently amended): The method as claimed in claim 4 ~~one of claims 4 to 14~~, in which step c) comprises the following operations:

c'0) an initial dictionary of initial resolution $\mathbf{r}_n$, higher than said given resolution $\mathbf{r}_N$, is obtained,

c'1) on the basis of the initial dictionary, an intermediate dictionary of resolution $\mathbf{r}_{n-1}$ lower than the initial resolution $\mathbf{r}_n$, is constructed by partitioning of the initial dictionary into several subsets ordered according to a predetermined criterion, and

c'2) operation c'1) is repeated until the given resolution $\mathbf{r}_N$ is attained.

Claim 19 (original): The method as claimed in claim 18, in which said predetermined criterion is chosen from among the cardinal of the subsets, an invoking of the subsets in a learning sequence, a contribution of the subsets to a total distortion or preferably to a decrease of this distortion.

Claim 20 (currently amended): The method as claimed in claim 18 ~~one of claims 18 and 19~~, in which said partition uses part at least of said insertion/deletion rules.

Claim 21 (currently amended): The method as claimed in claim 15 ~~claims 15 and 18~~, in which $\mathbf{N}$ successive dictionaries of respective resolutions $\mathbf{r}_1$ to $\mathbf{r}_N$ are obtained on the basis of an initial dictionary of intermediate resolution $\mathbf{r}_n$, by the repeated implementation of step c1) for the increasing resolutions $\mathbf{r}_{n+1}$ to $\mathbf{r}_N$, and through the repeated implementation of step c'1) for the decreasing resolutions $\mathbf{r}_{n-1}$ to $\mathbf{r}_1$.

Claim 22 (currently amended): The method as claimed in claim 4 ~~one of claims 4 to 21~~, in which, to adapt said dictionary to a use with a given dimension **N** of codevectors, steps a) and b), on the one hand, and step c), on the other hand, are substantially inverted so that:

- in step c), a first, intermediate, dictionary still of dimension **N'** but of higher/lower resolution $\mathbf{r}_N$ is constructed on the basis of an initial dictionary of resolution $\mathbf{r}_n$ and of dimension **N'** by embedding/simplification of dictionaries of increasing/decreasing resolutions, so as to substantially attain the resolution $\mathbf{r}_N$ of said first dictionary,
- in step a), to attain the given dimension **N**, a first set consisting of codevectors formed by inserting/deleting, into/from codevectors of the first dictionary of dimension **N'** lower/higher than said given dimension **N** elements taken from a finite set of real numbers according to a finite collection of predetermined insertion/deletion rules is constructed,
- and, in step b), subsequent to a possible step of definitive adaptation to the resolution $\mathbf{r}_N$, a second, definitive, dictionary comprising at least said first set is constructed for said given dimension **N**.

Claim 23 (currently amended): The method as claimed in claim 4 ~~one of claims 4 to 22~~, in which there is stored in a memory, once and for all, said collection of insertion/deletion rules, each identified by an index (l_r), and, for a given dimension:

- said second set consisting of codevectors that may not be obtained by application of the insertion/deletion to codevectors of lower/higher dimension than the given dimension according to said collection of insertion/deletion rules,
- as well as at least one correspondence table making it possible to reconstitute any codevector of the dictionary of given dimension, using the indices of the insertion/deletion rules and indices identifying elements of said second set,

thereby making it possible to avoid the complete storage of the dictionary for said given dimension, by simply storing the elements of said second set and links in the correspondence table for access to these elements and to the associated insertion/deletion rules.

Claim 24 (original): The method as claimed in claim 23, in which the correspondence tables are formulated previously, for each index (m^j) of a codevector (x^j) of the dictionary ($\mathbf{D}^j_{Nj}$) of given dimension (j) that may be reconstructed on the basis of elements of current indices (m') in the second set of current dimension (j'), through a tabulation of three integer scalar values representing:

- a current dimension (j') of said second set,
- a current index (m') of an element of the second set, and
- an insertion/deletion rule index (l_r),

this insertion/deletion rule at least contributing to reconstitute said codevector (x_j) of the dictionary (D^j_{Nj}) of given dimension (j), by applying the insertion/deletion to the element of said current index (m') and of said current dimension (j').

Claim 25 (currently amended): ~~A use of the dictionary obtained through the implementation of the method as claimed in one of claims 23 and 24, in the compression coding/decoding of digital signals, by vector quantization at variable rate defining a variable resolution~~ A method of using a dictionary in the compression coding/decoding of digital signals, by vector quantization at variable rate defining a variable resolution,

said dictionary, of a given dimension, giving codevectors reconstituted by using at least one correspondence table making it possible to reconstitute any codevector of the dictionary of said given dimension, using indices of a collection of insertion/deletion rules and indices identifying elements of a set of codevectors that may not be obtained by application of the insertion/deletion to codevectors of lower/higher dimension than the given dimension according to said collection of insertion/deletion rules,

in which a search is made for the codevector (x^j) which is the nearest neighbour of an input vector $y=(y_0, \ldots, y_k, \ldots, y_{j-1})$ in a dictionary (D^i_j) of given dimension (j),

and comprising the following steps:

CO1) for a current index (m^j) of said codevector (x^j) sought, reconstitution at least partial of a codevector of index (m') corresponding to said current index (m^j), at least through the prior reading of the indices (j', m', l_r) appearing in the correspondence tables making it possible to formulate said dictionary,

CO2) at least on coding, calculation of a distance between the input vector and the codevector reconstituted in step CO1),

CO3) at least on coding, repetition of steps CO1) and CO2), for all the current indices in said dictionary,

CO4) at least on coding, identification of the index (m_{min}) of the codevector at least partially reconstituted whose distance (d_{min}), calculated in the course of one of the iterations of step CO2), with the input vector is the smallest, and

CO5) at least on decoding, determination of the nearest neighbour of the input vector (y) in the guise of codevector (x^j) whose index (m_{min}) has been identified in step CO4).

Claim 26 (currently amended): The method ~~use~~ as claimed in claim 25, in which step CO1), at least on decoding, comprises:

CO11) the reading, in the correspondence tables, of indices representative of links to said second set and to the insertion/deletion rules and including:

- the index of a current dimension of a subset of said second set,
- the current index of an element of said subset,
- and the index of the appropriate insertion/deletion rule for the construction of the codevector of the dictionary of given dimension, on the basis of said element,

CO12) the reading, in the subset identified by its current dimension, of said element identified by its current index,

CO13) the complete reconstitution of the codevector to said given dimension by applying to said element read in step CO12) the appropriate insertion/deletion rule identified by its index read in step CO11).

Claim 27 (currently amended): The method ~~use~~ as claimed in claim 25, in which, on coding,

* step CO1) comprises:

CO11) the reading, in the correspondence tables, of indices representative of links to said second set and to the insertion/deletion rules and including:

- the index of a current dimension of a subset of said second set,
- the current index of an element of said subset,
- and the index of the appropriate insertion/deletion rule for the construction of the codevector of the dictionary of given dimension,

CO12) the reading, in the subset identified by its current dimension, of said element identified by its current index,

* in step CO2), said distance is calculated as a function of a distortion criterion estimated as a function of:

- the index of the insertion/deletion rule,
- and of the element of the subset identified by its current index,

thereby making it possible to only partially construct the codevector with said given dimension in step CO1), by reserving the complete reconstruction simply for decoding.

Claim 28 (currently amended): The method ~~use~~ as claimed in claim 25 ~~one of claims 25 to 27~~, in which there is provided furthermore a supplementary structuring property according to a union of permutation codes and utilizing an index of said union of permutation codes, and in which:

CP1) on the basis of an input signal, an input vector $y=(y_0, \ldots, y_k, \ldots, y_{j-1})$ defined by its absolute vector $|y|=(|y_0|,\ldots,|y_k|,\ldots,|y_{j-1}|)$ and by a sign vector $\varepsilon=(\varepsilon_0,\ldots,\varepsilon_k,\ldots,\varepsilon_{j-1})$ with $\varepsilon_k=\pm1$ is formed,

CP2) the components of the vector $|y|$ are ranked by decreasing values, by permutation, to obtain a leader vector $|\tilde{y}|$,

CP3) a nearest neighbour $x^{j'}$ of the leader vector $|\tilde{y}|$ is determined from among the leaders of the dictionary D^j_i of dimension j,

CP4) an index of the rank of said nearest neighbour $x^{j'}$ in the dictionary D^j_i is determined,

CP5) and an effective value of coding/decoding is applied to the input vector, which is dependent on said index determined in step CP4), on said permutation determined in step CP2) and on said sign vector determined in step CP1).

Claim 29 (currently amended): The method ~~use~~ as claimed in claim 25 ~~one of claims 25 to 28~~, in which at least said correspondence tables are stored in a memory of a coding/decoding device.

Claim 30 (currently amended): A computer program product intended to be stored in a memory of a processing unit, in particular of a computer or of a mobile terminal, or on a removable memory medium and intended to cooperate with a reader of the processing unit,

~~characterized in that it comprises instructions for the implementation of the method as claimed in one of claims 4 to 24~~ wherein it comprises instructions for implementing a method for forming a dictionary comprising codevectors of variable dimension and intended to be used in a device for compression coding and/or decoding of digital signals, by vector quantization at variable rate defining a variable resolution,
in which, for a given dimension:
a) a first set consisting of codevectors formed by inserting/deleting into/from codevectors of dictionaries of lower/higher dimension elements taken from a finite set of real numbers according to a finite collection of predetermined insertion/deletion rules,
b) a first, intermediate, dictionary comprising at least said first set is constructed, for said given dimension,
c) and, to adapt said dictionary to a use with at least one given resolution, a second, definitive, dictionary is constructed, on the basis of the intermediate dictionary, by embedding/simplification of dictionaries of increasing/decreasing resolutions, the dictionaries of increasing resolutions being inter-embedded from the dictionary of smallest resolution up to the dictionary of greatest resolution.

Claim 31 (currently amended): A computer program product intended to be stored in a memory of a processing unit, in particular of a computer or of a mobile terminal integrating a coding/decoding device, or on a removable memory medium and intended to cooperate with a reader of the processing unit,
~~characterized in that it comprises instructions for the implementation of the application to compression coding/decoding as claimed in one of claims 25 to 29~~ wherein it comprises instructions for implementing a use of a dictionary in the compression coding/decoding of digital signals, by vector quantization at variable rate defining a variable resolution,
said dictionary, of a given dimension, giving codevectors reconstituted by using at least one correspondence table making it possible to reconstitute any codevector of the dictionary of said given dimension, using indices of a collection of insertion/deletion rules and indices identifying elements of a set of codevectors that may not be obtained by application of the insertion/deletion to codevectors of lower/higher dimension than the given dimension according to said collection of insertion/deletion rules,

in which a search is made for the codevector (x^j) which is the nearest neighbour of an input vector $y=(y_0, \ldots, y_k, \ldots, y_{j-1})$ in a dictionary (D^i_j) of given dimension (j),

and comprising the following steps:

CO1) for a current index (m^j) of said codevector (x^j) sought, reconstitution at least partial of a codevector of index (m') corresponding to said current index (m^j), at least through the prior reading of the indices (j', m', l_r) appearing in the correspondence tables making it possible to formulate said dictionary,

CO2) at least on coding, calculation of a distance between the input vector and the codevector reconstituted in step CO1),

CO3) at least on coding, repetition of steps CO1) and CO2), for all the current indices in said dictionary,

CO4) at least on coding, identification of the index (m_{min}) of the codevector at least partially reconstituted whose distance (d_{min}), calculated in the course of one of the iterations of step CO2), with the input vector is the smallest, and

CO5) at least on decoding, determination of the nearest neighbour of the input vector (y) in the guise of codevector (x^j) whose index (m_{min}) has been identified in step CO4).